PCIe® 4.0 Protocol Update

Steve Glaser

PWG Member

NVIDIA

Disclaimer

The information in this presentation refers to specifications still in the development process. This presentation reflects the current thinking of various PCI-SIG® workgroups, but all material is subject to change before the specifications are released.

Agenda

- **Completed ECNs Against the 3.1/3.1a Base Specs**
  - Emergency Power Reduction with PWRBRK Signal
  - RC Integrated Endpoint & IOV Updates
  - Expanded Resizable BARs
  - Extended Message Data for MSI
  - SR-IOV Table Updates
  - VF Resizable BARs
  - Flattening Portal Bridge (FPB)
  - Hierarchy ID Message
  - Expansion ROM Validation
  - Native PCIe Enclosure Management (NPEM)
- **Major Protocol Spec Changes for PCIe 4.0**

Emergency Power Reduction with PWRBRK Signal ECN

Emergency Power Reduction with PWRBRK Signal – Overview

- **Hardware-based mechanism to rapidly reduce consumed power under emergency situations**
  - Main intent is to avoid system damage, not manage power
  - The PWRBRK signal overrides other power mgmt mechanisms
- **Three sets of related changes**
  - Card Electromechanical (CEM) spec updated to allow use of Rsvd pin B30 for the new PWRBRK signal
  - PCIe Base spec updated with associated control & status bits
  - SR-IOV spec updated with associated control & status bits
- **Software implications**
  - No SW changes required to use the new mechanism
  - System SW can optionally use the new control & status bits to confirm correct operation of the mechanism

Emergency Power Reduction (EPR) with PWRBRK Signal – Details

- **Except with system board devices, Functions supporting this mech must report Power Budgeting values for this state**
- **This state is associated with a Device. All Functions in a Device that support it enter and exit this state at the same time.**
- **Functions may require re-initialization upon exit from this state**
- **While in this state, Functions may continue to operate in a degraded mode, or may direct their Upstream Port to DL_Down**
- **For Switches, Downstream Switch Ports enter and exit this state at the same time as the associated Upstream Switch Port**
- **For SR-IOV Devices, VFs enter and exit this state at the same time as their PF**
- **The EPR Detected bit permits SW to detect that this state was entered, even momentarily**
- **SW can direct a component to enter this state using the EPR Request bit**

RC Integrated Endpoint & IOV Updates ECN

RC Integrated Endpoint (RCiEP) & IOV Updates – Overview

- **Combined with associated errata, this ECN implements a variety of changes to support more consistent implementations of RCiEPs, notably wrt SR-IOV**
- **Key sets of changes**
  - **Alternative Routing Interpretation:** ARI is not applicable to RCiEPs, and RCiEPs have special rules for VF number assignment
  - **Access Control Services:** Implementing ACS with RCiEPs is permitted but not required, and some ACS rules are different
  - **Process Address Space ID:** A new ImpNote provides guidance for the PASID width in systems containing RCiEPs
  - **Device Serial Number:** With RCiEPs, some rules are different
  - **Latency Tolerance Reporting:** With MFD RCiEPs, multiple RCiEPs are permitted to implement the LTR capability

Expanded Resizable BARs ECN

Expanded Resizable BARs Background & Motivation

- **Original Resizable BARs ECN was released in 2008**
  - Mechanism to support BAR size negotiation
  - Typical HW default is a modest-sized BAR setting
  - After boot, resizable-aware SW can select a different size
  - 20 architected sizes, from 2^{20} (1 MB) to 2^{39} (512 GB) in powers of 2
- **New demands for larger-sized BARs**
  - Large amounts (>512 GB) of DRAM or non-volatile memory (NVM)
    - Endpoints capable of 2 TB are already shipping
    - Emerging NVM technologies promise densities >10x that of flash
  - Some system architectures now using PCIe Memory Space as a virtualized address space
    - x86_64 supports 48b; ARM supports 49b; PPC supports 52b
- **The Expanded Resizable BARs ECN adds more architected sizes in a backwards-compatible manner**

Expanded Resizable BARs Details

- 20 additional sizes, from 2^{40} (1 TB) to 2^{63} (8 EB) in powers of 2
- Backwards compatible with SW based on the original ECN

Resizable BAR Capability Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECN's BAR supported sizes								Pre-ECN BAR supported sizes																				RsvdP			

Resizable BAR Control Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECN's BAR supported sizes																RsvdP			BAR Size					# BARs			RsvdP			BAR Idx	

- New Resizable BAR Capability bits indicate if device supports operating with the larger sizes
- Resizable BAR Control Register's BAR size field increases by one bit to enable indicating the larger sizes

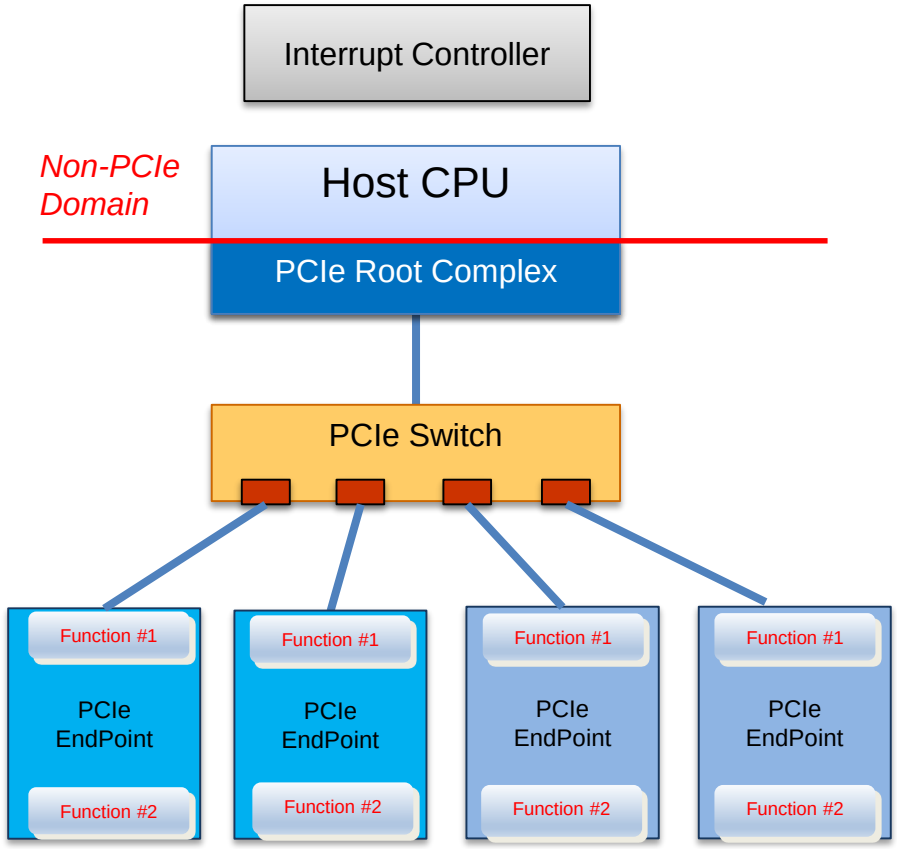
Extended Message Data for MSI ECN

Extended Message Data for MSI Overview

- **Enables an MSI to support a 16-bit Extended Message Data value in addition to the 16-bit Message Data Value already supported**
- **This provides MSI with a similar capability to MSI-X, which passes 32 bits of Message Data**
- **Useful for mobile and other devices that do not typically implement MSI-X**

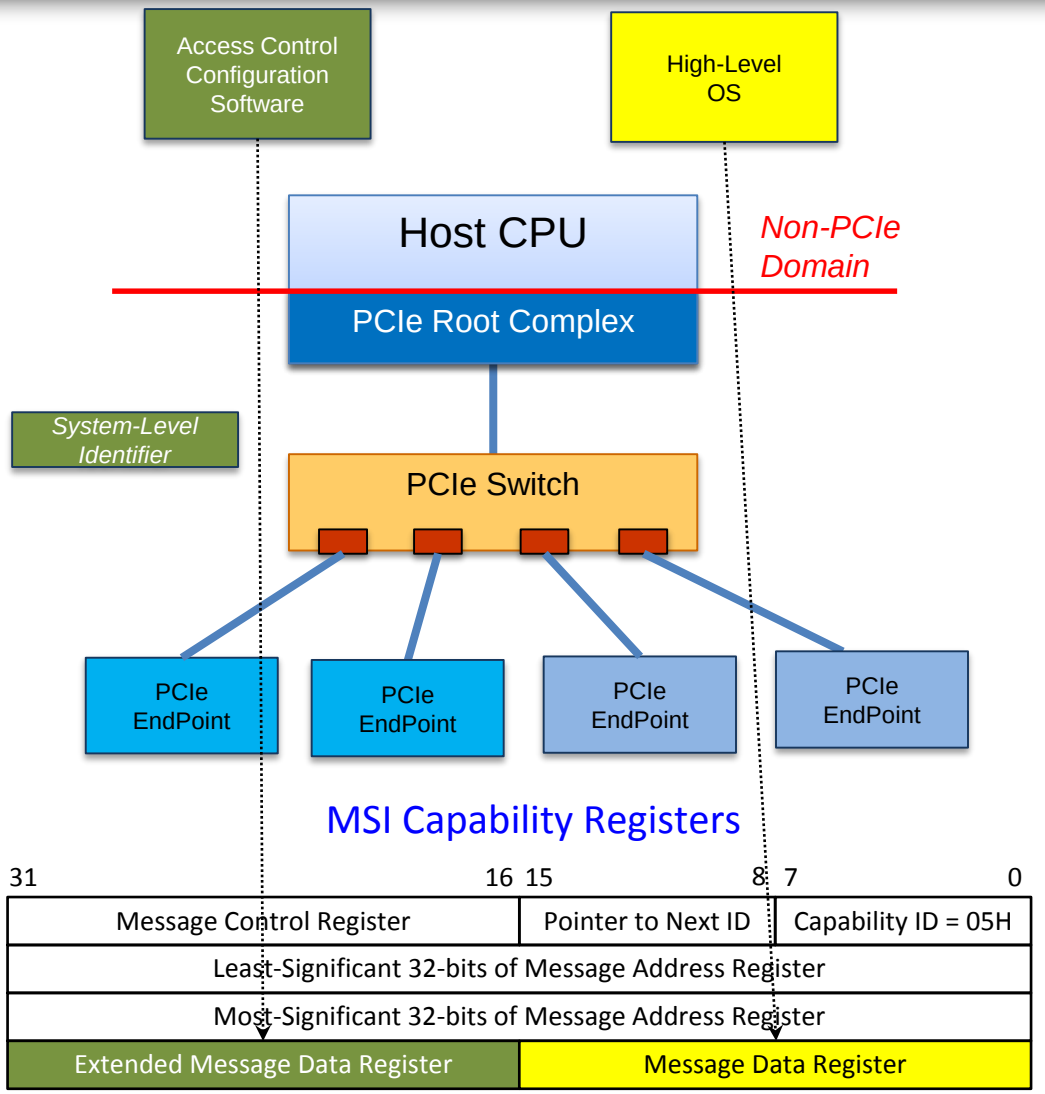
Extended Message Data for MSI Motivation

- **Interrupt priority or other useful information may be provided in Extended Message Data**
- **For example, in mobile devices, Extended Message Data may be used to pass in-band a "System-Level Identifier" (SLI) of the Endpoint Function that generated the MSI**
- **An Interrupt Controller may perform access control using the SLI to determine if the MSI is permitted to deliver an interrupt to a specific Memory address**



Extended Message Data for MSI Detailed Example

- **A capability exists in many SoC designs to set "per register" access rights for different host software entities**
- **Access Control is performed using:**
  - MSI Data
  - MSI Address
  - Extended Message Data (System-Level Identifier)
- **Access Control Configuration SW is permitted to write the System-Level Identifier to the Extended Message Data Register**
- **The High-Level OS is permitted to write into the Message Data Register**



SR-IOV Table Updates ECN

SR-IOV Table Updates

- **Primarily updates the SR-IOV spec to reflect more recent PCI Capabilities and PCIe Extended Capabilities**
  - Also includes small updates to the Readiness Notifications (RN) ECN
  - Some preparation for incorporating SR-IOV into the 4.0 Base spec
  - Many other changes will be made as a part of the incorporation
- **Table 3-21: SR-IOV Usage of PCI Standard Capabilities**
  - Added entries for Null & Enhanced Allocation
  - SATA capability now permitted for PFs
- **Table 3-22: SR-IOV Usage of PCI Express® Extended Capabilities**
  - Ten new entries for more recent Extended Capabilities
  - Clarified that VFs are forbidden from implementing certain Extended Caps
  - Several corrections to Extended Capability descriptions
- **Other Updates**
  - Device Serial Number: rules & recommendations for PFs vs VFs
  - Power Budgeting: permitted in PFs, but not VFs
  - Resizable BAR: permitted in PFs, but not VFs; separate ECN for VF RBARs
  - Process Address Space ID: PASID permitted in PFs, but not VFs; other rules
  - Readiness Time Reporting: permitted in PFs and/or VFs; other rules

VF Resizable BARs ECN

VF Resizable BARs

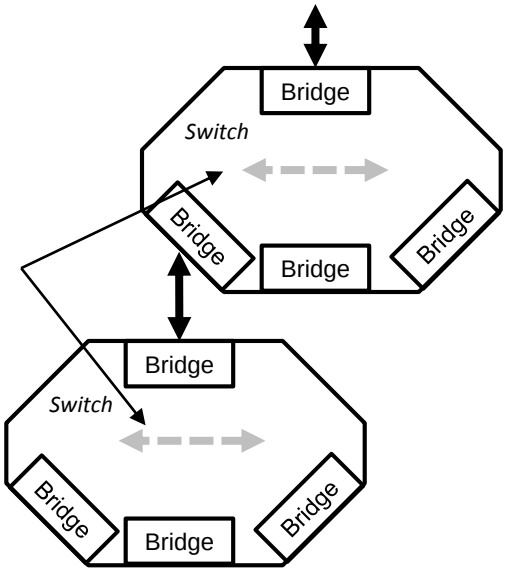
- **ECN is against the SR-IOV spec**
- **Similar to the combination of the (old) RBAR & (new) Expanded RBAR ECNs**
- **Adds the capability to resize VF BARs in PFs**
  - Seriously investigated adding this to the existing RBAR ECNs, but this approach would not have been backwards compatible with existing RBAR SW
  - Is a clean solution that simply defines a new Extended Capability explicitly for this purpose
  - Much of the new text is highly leveraged from the existing RBAR ECNs
  - Some of the new register descriptions simply refer to existing RBAR register descriptions

Flattening Portal Bridge (FPB) ECN

Flattening Portal Bridge (FPB) Overview

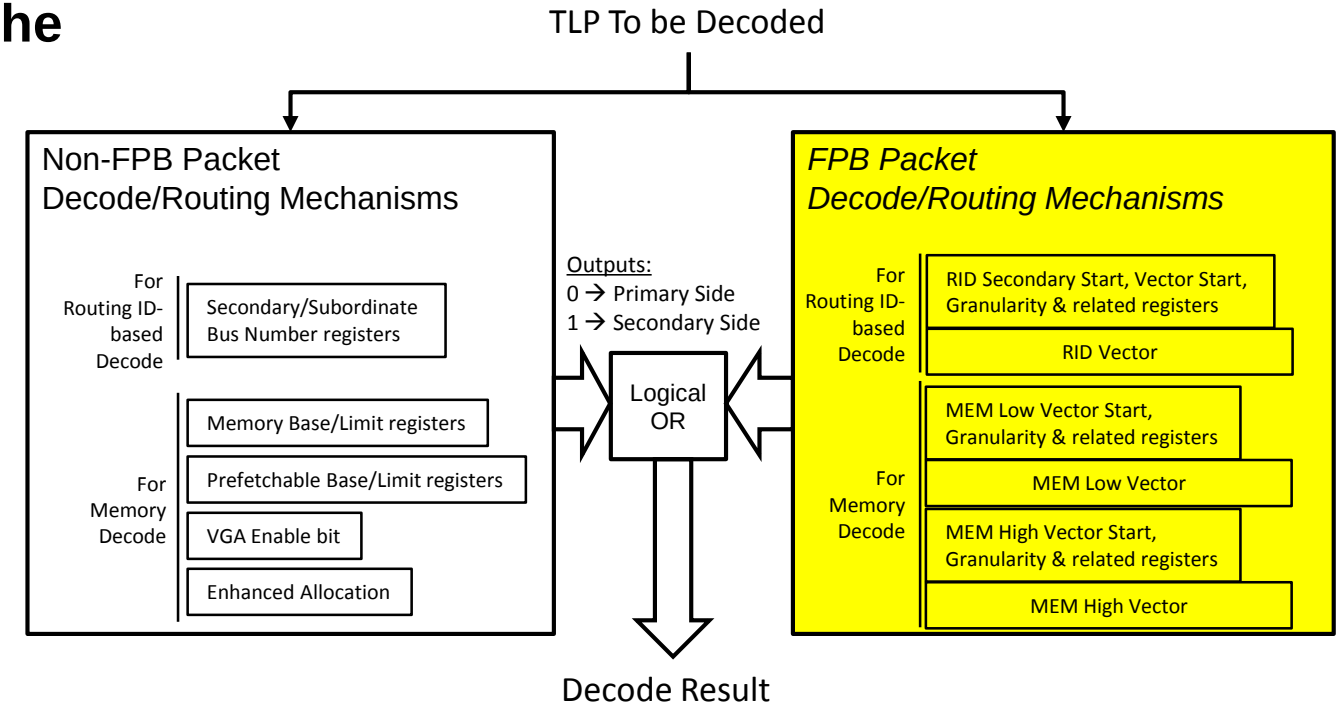
- **FPB improves MMIO and Config Space allocation efficiency**
- **“Flattens” the use of Routing IDs in Switches to make more efficient use of available space**
- **Avoids rebalancing by enabling noncontiguous resource range (re/)allocation for both Routing IDs and MMIO**

When enabled, FPB changes the Switch internal structure to remove the logical internal bus, allowing the Downstream Ports to directly follow the Upstream Port in BDF space



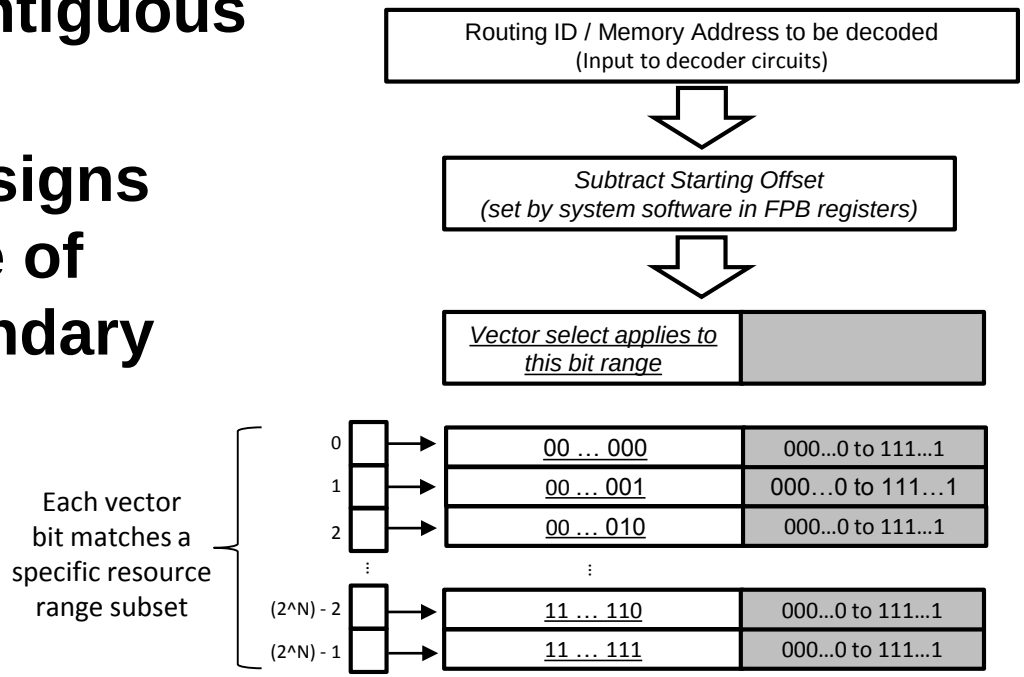
Flattening Portal Bridge (FPB) Relationship with Existing Mech

- **Each Bridge with FPB has both the existing and new (yellow) mechanisms**
- **Software has option to use either or both**
- **Hardware decode result is the "OR" of the two**



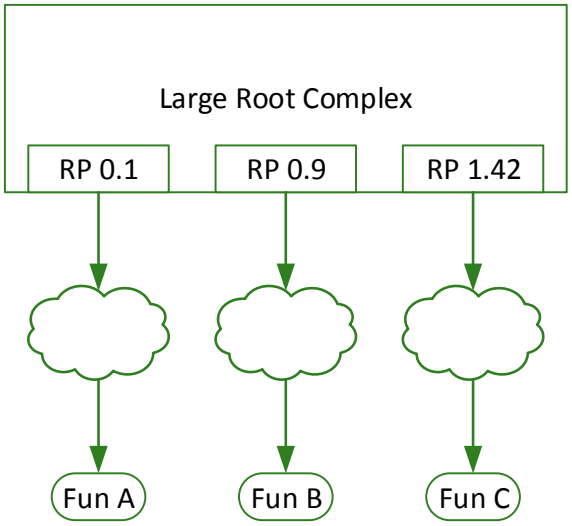
Flattening Portal Bridge (FPB) Noncontiguous Resource Decode

- **FPB defines a bit vector mechanism for noncontiguous resource decode**
- **Each bit, when Set, assigns a corresponding range of resources to the Secondary Side of the bridge**
- **There are registers to assign the starting offset and size of the range a bit corresponds to**



Hierarchy ID Message ECN

Hierarchy ID Problem



RP 0.x is one Hierarchy
RP 1.y is another Hierarchy

- **Bus enumeration assigns Routing IDs (RIDs) to Functions**
  - RID is unique only within a Hierarchy
- **Functions don't know which Hierarchy they are part of**
  - RID + Hierarchy ID + System ID is globally unique
  - Functions in the same Hierarchy can communicate over PCIe
- **Interesting Topologies:**
  - Large Root Complexes
  - Clustered Systems
  - Fault Tolerant Systems

Hierarchy ID Solution

- **New Type 1 Vendor Defined Message (VDM)**
  - Optional, Broadcast
  - Type 1 VDMs must be ignored by existing receivers that don't support them
- **Send from Downstream Port (ideally Root Port)**
  - Software says when to send HID Message and what HID Message contains
  - Hierarchy ID is the "Segment Group Number" defined in the Firmware Spec
  - System GUID can use a variety of schemes (IEEE UID-64, Vendor Serial #, …)
- **Received by Upstream Ports**
  - Reported in capability for software/driver/firmware
- **Supports Root Complex Integrated Endpoints (RCiEP)**
  - No message on the wire – software writes RCiEP Hierarchy ID Capability
- **No operational effects**
  - Provides information to hardware / software / …
  - Does not otherwise affect PCIe operation

- **Comparison to Device Serial Number (DSN):**
  - DSN provides unique number, but usually needs a ROM
  - DSN doesn't provide location information (where am I attached)

Expansion ROM Validation ECN

Expansion ROM Validation

- Implementation-specific methods to support expansion ROM validation. This ECR does not affect those.
- This ECN defines a standardized mechanism to report validation results.
  - The report is advisory
  - Does not affect access to the ROM
- ECN defines a new 3-bit field in the Expansion ROM BAR to report status
  - 8 encodings of Pass/Fail/InProgress/NotSupported status
- Software is then able to take appropriate action
  - Permits the ROM contents to be used or not
  - Contains errors/adapter

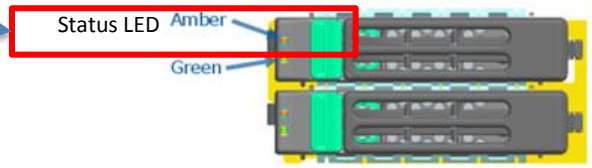
Native PCIe Enclosure Mgmt (NPEM) ECN

History and Motivation for NPEM

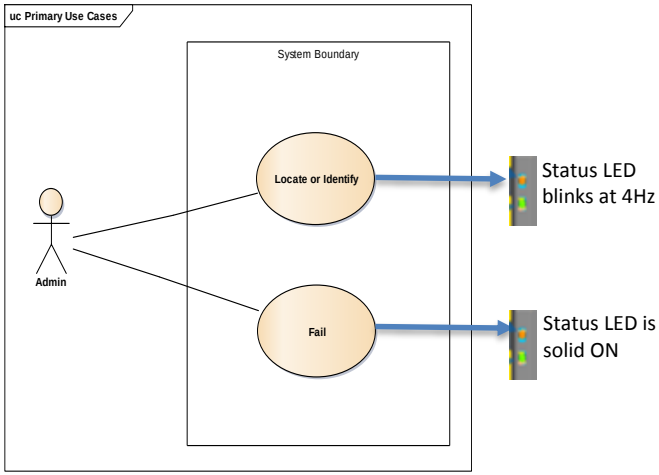
- **Disk arrays (e.g., RAID) require a visual indication of drive status (failed, critical, rebuilding, locate, etc.)**
  - Without such a visual indication, for instance, an admin can erroneously remove a good disk from the critical (degraded) RAID5 array in place of a failed disk that could lead to data loss.
- **Storage has established enclosure/LED models not architecturally supported in PCIe today**
  - SFF-8489 standard defines blink patterns for storage in accordance with the International Blinking Pattern Interpretation (IBPI).
  - SAS/SATA ecosystem defines a simple SGPIO interface for simple enclosure management (e.g., LED control)
  - A simple SGPIO unit is typically integrated to a central SAS/SATA controller/HBA.
- **Enclosure/LED function is under the purview of PCIe**
  - Unlike SAS/SATA, In NVMe, the controller/HBA is part of each drive thus the notion of a separate central controller is eliminated.
  - In typical white box server implementations for NVMe, enclosure function is inside a root port or a downstream switch port to which NVMe drive is connected. This takes the enclosure function outside of the NVMe subsystem and brings it under the purview of PCIe.
- **The NPEM proposal was brought into the PWG by the NVMe Storage Community in the NVMe-MI workgroup**

NPEM defines mechanisms for storage enclosure management for NVMe SSDs, consistent with established capabilities in the storage ecosystem

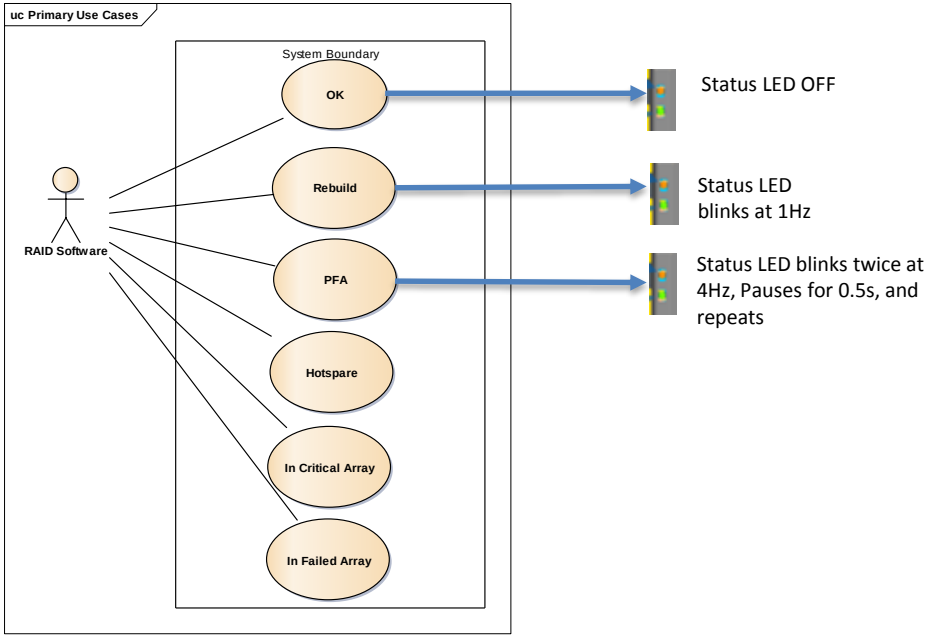
NPEM (Storage LED) Use Cases



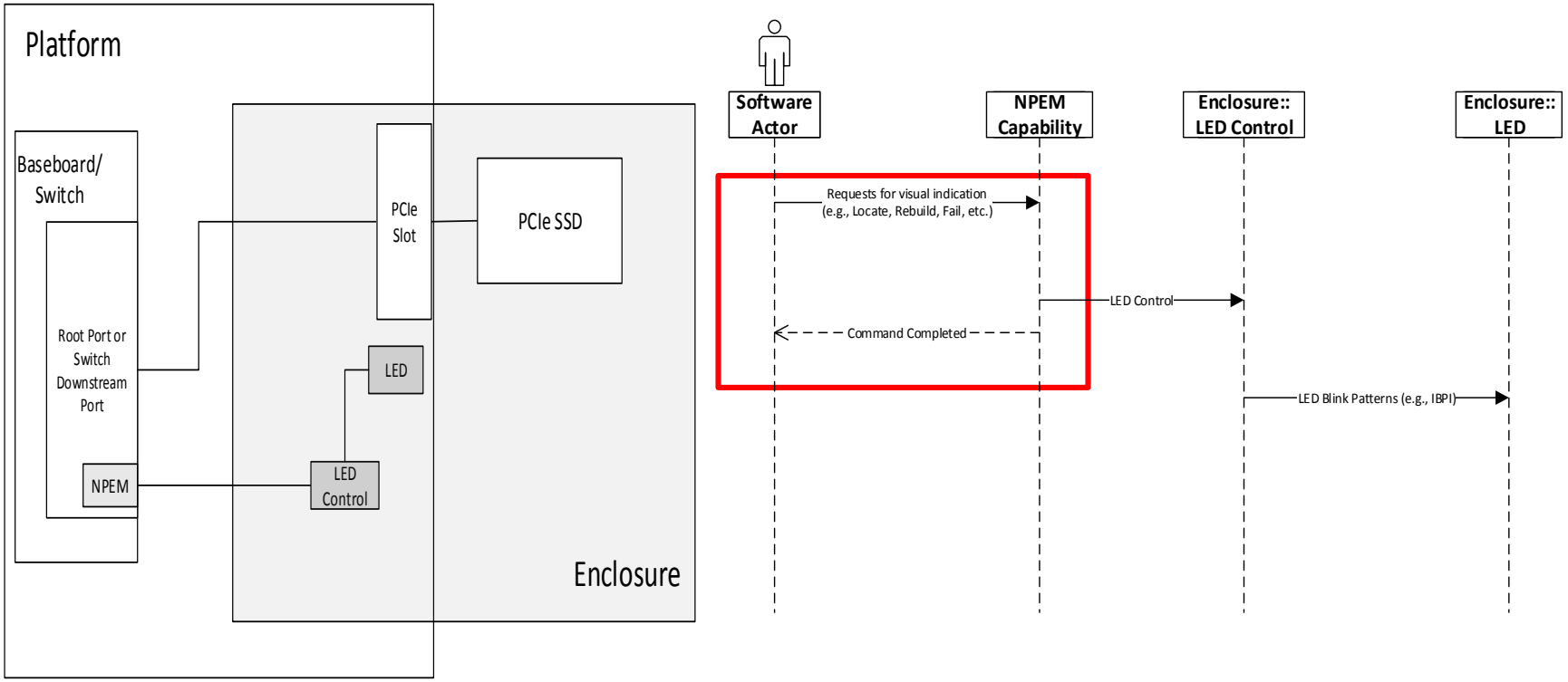
Admin initiates GUI action



RAID Software initiates action

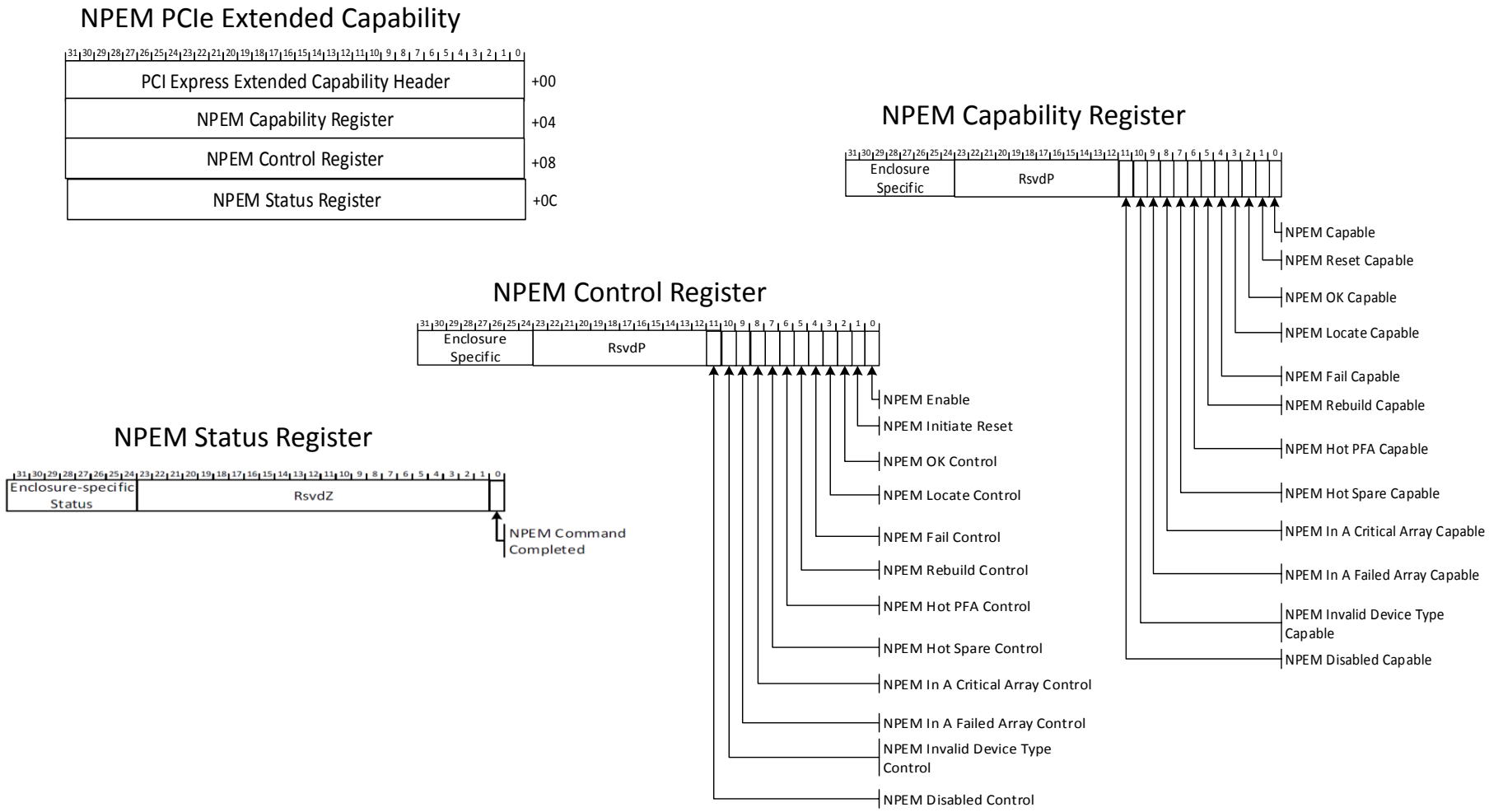


NPEM System View



NPEM is an optional PCIe Extended Capability that provides mechanisms for enclosure management. This mechanism is designed to provide management for enclosures containing PCIe SSDs that is consistent with the established capabilities in the storage ecosystem.

NPEM PCIe Extended Capability



Major Protocol Spec Changes For PCIe 4.0

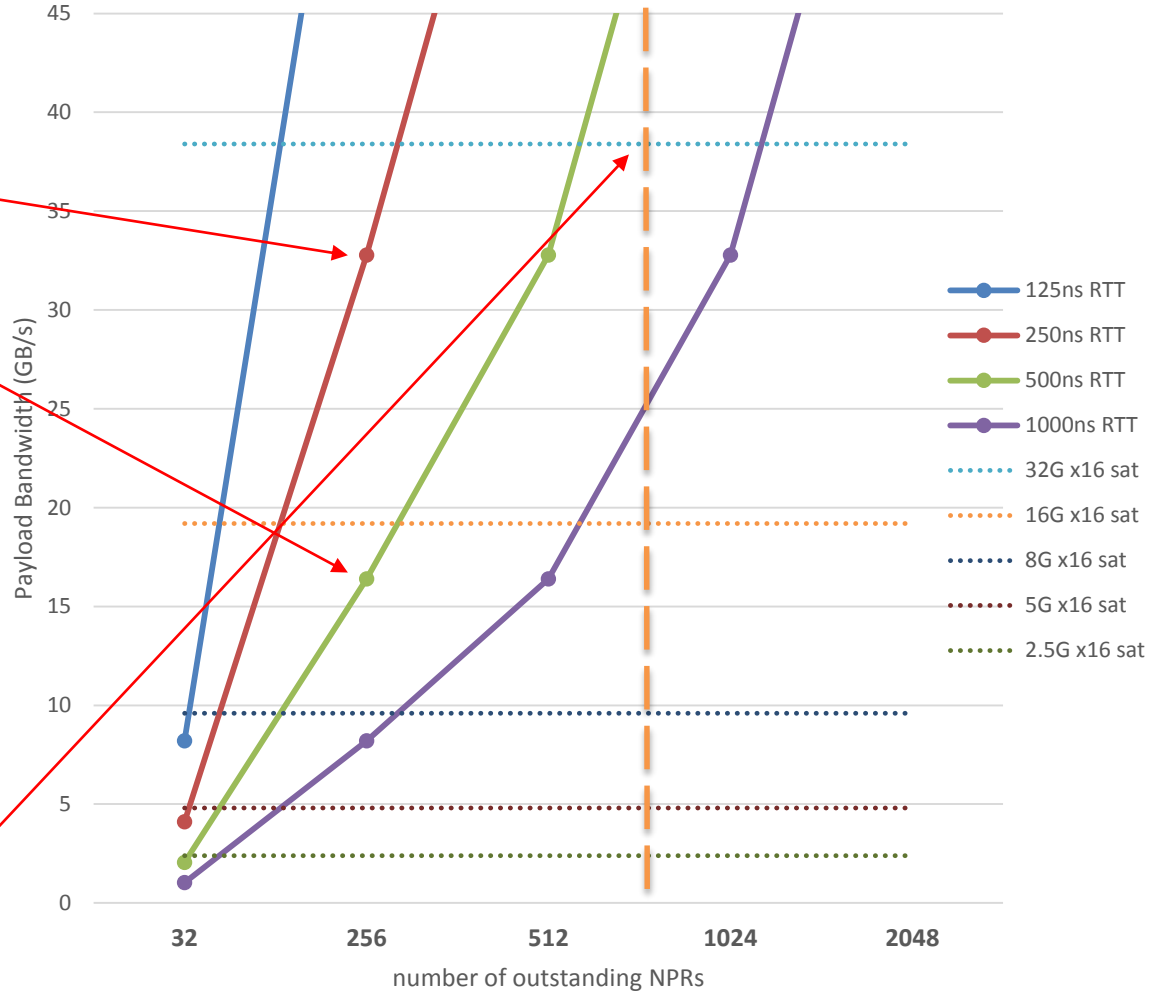
- 10-Bit Tags
- Scaled Flow Control
- Simplified Protocol Timers
- Other PCI Spec Integration

4.0 Changes: 10-Bit Tags Problem Setup

- **Max of 256 outstanding Non-Posted Requests due to 8-bit Tag field**
- **Some workloads are demanding higher numbers**
  - One important class is GPUs or GP-GPU accelerators
  - High bandwidth with relatively small transaction sizes
  - Will use 32-byte transaction size for these calculations
- **Basic formula: BW = S * N / RTT with non-saturated Links, where:**
  - BW = payload bandwidth
  - S = transaction payload size
  - N = number of outstanding Non-Posted Requests (NPRs)
  - RTT = transaction round-trip time
- **Ability to accommodate larger RTTs is highly desirable:**
  - Switch and/or Retimer topologies
  - Longer latencies due to heavy system loading
  - Higher host memory latencies with larger systems

4.0 Changes: 10-Bit Tags Payload BW For 32-Byte NPRs

- **"Sat" (saturated) Link payload capacity based on 60% of encoded BW**
- **For 256 NPRs with 250ns RTT, not able to saturate a future 32G x16 Link**
- **For 256 NPRs with 500ns RTT, not able to saturate a 16G x16 Link**
- **Higher # of outstanding NPRs needed to accommodate longer RTTs, faster Links, and/or smaller payload NPRs**
- **10-Bit Tags effectively supports an additional "1.5" Tag field bits, corresponding to 768 NPRs**
- **768 NPRs with 500ns RTT can saturate a 32G x16 Link**



4.0 Changes: Scaled Flow Control Problem Statement

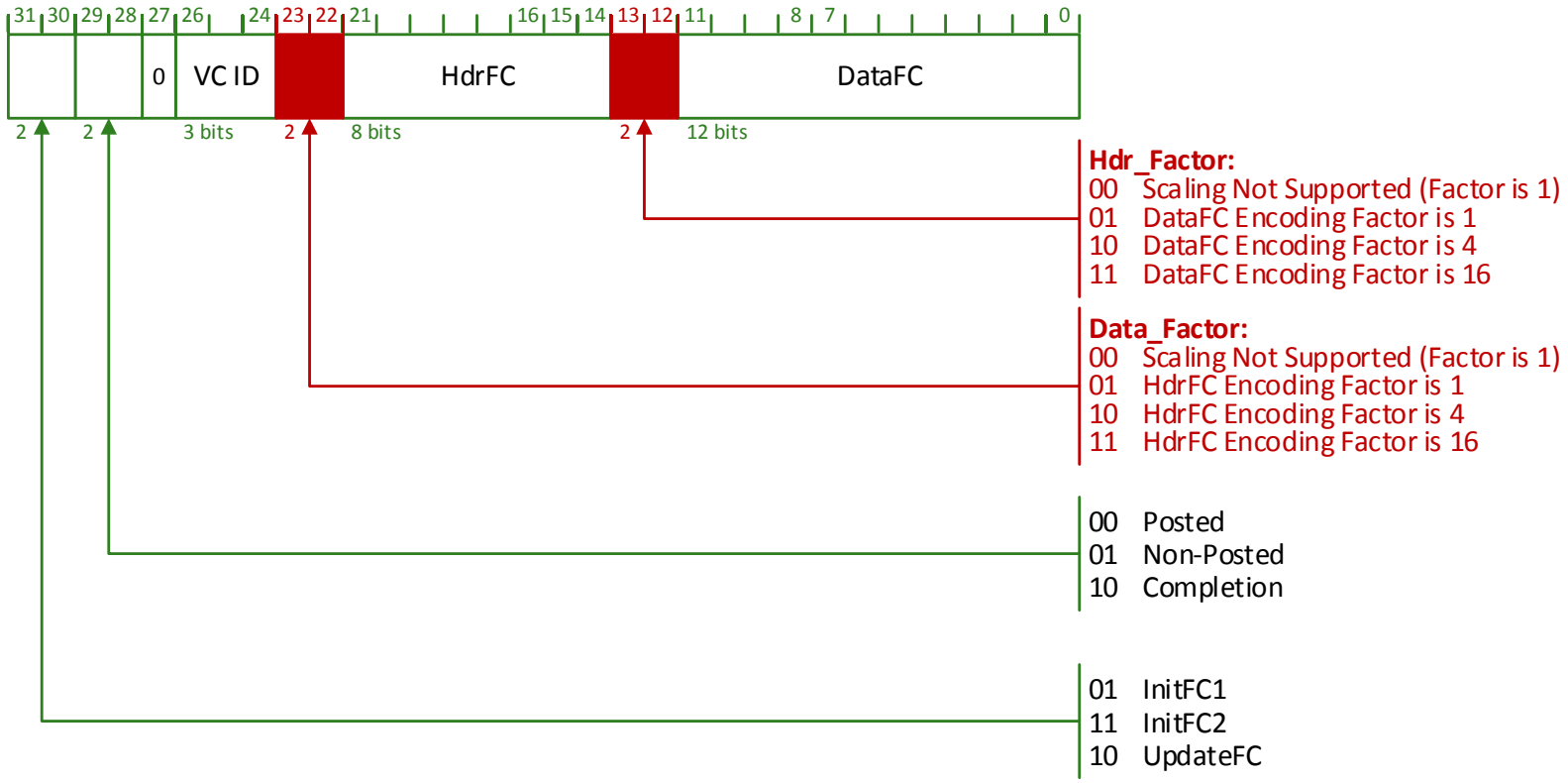
- **16GT/s bandwidth is higher**
- **16GT/s latency is mostly unchanged**
- **Existing platforms are running into limits**
  - Up to 127 Outstanding Header Credits
  - Up to 2047 Outstanding Data Credits
  - 8GT/s x16 can't deliver full Link bandwidth in some situations
- **Flow Control is independent of Link Speed**
  - Link can train to speed X and then switch to speed Y without renegotiating flow control
  - Don't tie max outstanding credits to current Link Speed

4.0 Changes: Scaled Flow Control Widen Internal Fields, Send Upper Bits

	Support	Encoding Factor	HdrFC	DLLP HdrFC:	DataFC	DLLP DataFC:
0	No	1	8 bits	HdrFC[7:0]	12 bits	DataFC[11:0]
1	Yes	1	8 bits	HdrFC[7:0]	12 bits	DataFC[11:0]
2	Yes	4	10 bits	HdrFC[9:2]	14 bits	DataFC[13:2]
3	Yes	16	12 bits	HdrFC[11:4]	16 bits	DataFC[15:4]

- **Credits don't change**
  - 1 Header Credit remains 1 TLP Header
  - 1 Data Credit remains 16 bytes
- **Support is per Link**
  - Negotiated using new Data Link Feature DLLP
- **Encoding Factor varies by credit pool**
  - {Posted, Non-Posted, Completion} × {Header, Data}
  - Receiver selects; Transmitter uses what Receiver specifies

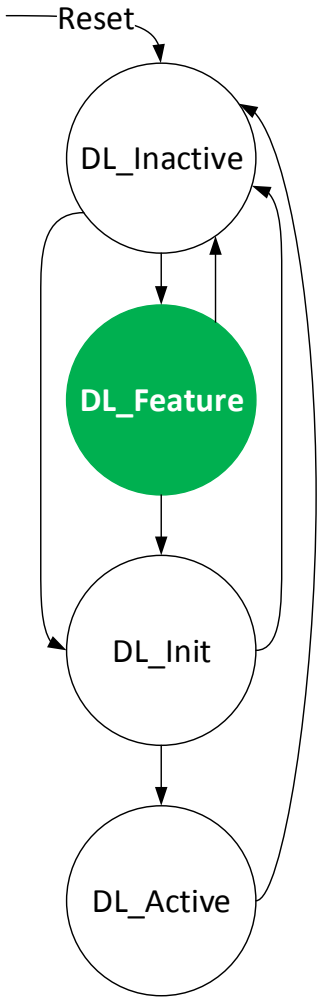
4.0 Changes: Scaled Flow Control New DLLP Content



- **Encoding Factor reflects Receiver's input buffer**
- **Support reflects your Receiver and Transmitter**

4.0 Changes: Scaled Flow Control DL_Feature state

- **Indicates support for Scaled Flow Control**
- **Send Data Link Feature DLLP (DLF)**
  - Feature Ack bit – for handshake, initially 0b
  - Feature bits
    - Bit 0 – Scaled Flow Control Supported
    - Others reserved for future features
- **If Receive DLF**
  - Record feature bits in new Capability structure
  - Set Feature Ack in transmitted DLF
- **Exit to DL_Init if:**
  - Receive DLF with Feature Ack set
  - Receive InitFC1
- **Scaled Flow Control enabled only if both Ports support it**



4.0 Changes: Simplified Protocol Timers

- **Problem: various protocol timers had separate tables for each Link speed; i.e., one table each for 2.5G, 5G, & 8G**
  - UPDATEFC TRANSMISSION LATENCY GUIDELINES
  - UNADJUSTED REPLAY_TIMER LIMITS
  - ACK TRANSMISSION LATENCY LIMIT AND ACKFACTOR
  - Didn't want to add yet another set of tables for 16G
- **Explored various options to permit new implementations to simplify their timer logic, while still backwards compatible**
  - Replay Timer: explored changing % tolerances to max numbers of Symbol Times, and dramatically simplifying the formula
  - Explored components negotiating Replay Timer values based on their Replay Buffer sizes
  - Considered SW override mechs for various timer values, to enable enhanced debugging for timer-related issues

4.0 Changes: Integrating Other PCI Specifications

- **Problems**
  - The PCIe Base spec was written assuming that readers have significant knowledge of PCI Local Bus specs, which is becoming less & less true
  - Some key PCIe functionality is specified only in earlier PCI Local Bus specs, which use different terminology & documentation conventions
  - A number of PCI specs are no longer being maintained
- **Solution: integrate key PCI specs into the PCIe 4.0 Base spec**
- **Specs that were integrated:**
  - *PCI™ Local Bus Specification, Revision 3.0*
  - *PCI Bus Power Management Interface Specification, Revision 1.2*
  - *Address Translation Services, Revision 1.1*
  - *Single Root I/O Virtualization and Sharing Specification Revision 1.1*

Thank you for attending the PCI-SIG Developers Conference Asia-Pacific Tour 2017.

For more information please go to www.pcisig.com